TECHNICAL REPORT CAMS–98.9.2

Center for Applied Mathematical Sciences

Denney Research Building 308
University of Southern California
Los Angeles, CA 90089-1113
Telephone: (213) 740-2395

DOMAIN PURSUIT METHOD FOR TRACKING BALLISTIC TARGETS

TECHNICAL REPORT CAMS–98.9.2

CHUANXIA RAO, BORIS ROZOVSKY AND ALEXANDER TARTAKOVSKY

CENTER FOR APPLIED MATHEMATICAL SCIENCES
University of Southern California
Los Angeles, CA 90089–1113

SEPTEMBER 1998

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
9-2-98

3. REPORT TYPE AND DATES COVERED
Technical

4. TITLE AND SUBTITLE
DOMAIN PURSUIT METHOD FOR TRACKING BALLISTIC TARGETS

5. FUNDING NUMBERS
DAAG55-98-1-0418

6. AUTHOR(S)
Chuanxia Rao, Boris Rozovsky, and Alexander Tartakovsky

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)
Center for Applied Mathematical Sciences
University of Southern California
Los Angeles, CA 90089-1113

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
ARO 38076.2-MA

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

In this report a fast optimal nonlinear filter is developed. This fast filter is based on two techniques: (I) splitting of the convection and diffusion operators and (II) tracking of the "important" domains (windows) with iterative reduction of the window size – a domain pursuit technique. As a result, the domain of interest is determined adaptively. The developed nonlinear filtering algorithm is then applied to a real problem of tracking ballistic targets in six dimensions. Simulation results for this problem demonstrate the fairly high statistical accuracy, efficiency, and real-time performance of the proposed algorithm. These results also show that the developed method is much more accurate compared to the traditional extended Kalman filter.

14. SUBJECT TERMS
Nonlinear Filtering, Domain Pursuit Method, Tracking of Ballistic Missiles

15. NUMBER IF PAGES
19

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

ENCLOSURE 1

Contents

Abstract

In this report a fast optimal nonlinear filter is developed. This fast filter is based on two techniques: (i) splitting of the convection and diffusion operators and (ii) tracking of the "important" domains (windows) with iterative reduction of the window size – a domain pursuit technique. As a result, the domain of interest is determined adaptively.

The developed nonlinear filtering algorithm is then applied to a real problem of tracking ballistic targets in six dimensions. Simulation results for this problem demonstrate the fairly high statistical accuracy, efficiency, and real-time performance of the proposed algorithm. These results also show that the developed method is much more accurate compared to the traditional extended Kalman filter.

1 Introduction

In applications the target tracking problem can be naturally formulated as a filtering problem for hidden Markov models: given an unobserved (hidden) state (or signal or system) process, which is usually assumed Markovian, and an observation (or measurement) process, which provides noisy information about the state process, one needs to estimate the state or a function of the state at a given time moment by using all the observational information available up to that time moment.

Ideally, the involved stochastic dynamics are linear and Gaussian and in this case the filtering problem is solved by the Kalman and Kalman-Bucy filters. For linear and Gaussian models the Kalman filter is optimal in the mean-square sense and had a big success in a wide variety of applications. However, many real-world problems do not fit well with linear dynamic models. Sometimes one can explicitly describe the distribution of the state given measurements (posterior distribution) but, outside the realm of the linear theory, only a very few examples have explicitly described posterior distributions. Since most real problems are nonlinear, this creates a fundamental problem. Successive linearization in short time intervals, the Extended Kalman Filtering procedure, may be applied but its serious disadvantage is that it often gives erroneous answers and refining of computational effort can increase them.

Theoretical study of the general nonlinear filtering problem has also gone through more than three decades of efforts of mathematicians, statisticians, and engineers and has now become more or less mature as a research field. See the books by Stratonovich [28], Jazwinski [11], Liptser and Shiryayev [16], Kallianpur [13], Rozovskii [26], Pardoux [22], Bensoussan [2], Tanizaki [29], Elliott, Aggoun, and Moore [4]. In contrast to the linear case where there exists a finite dimensional statistic, a general nonlinear problem is infinite-dimensional in nature. This was the main reason why optimal nonlinear filters have not been widely used in real applications.

Approximations to the optimal nonlinear filter must be adopted. The simplest and most widely used approximation is the extended Kalman filter (EKF), which is basically the Kalman-Bucy filter applied to a dynamically linearized system. The EKF has been modified for different purposes and has many versions such as the second order approximation [21] and the iterative extended Kalman filter [1, 11, 12].

In addition to the EKF and its modifications, there are two major approaches to nonlinear filter approximation. One approach assumes that the filtering densities belong to a certain class of functions such as Gaussian or exponential or some combinations. Actually EKF is an example of an assumed-density filter and is perhaps the simplest. Other examples are the Gaussian mixture filter (see [27]) and more recently the projection filter (see [3]).

The other approach is to use a direct (analytical or numerical) approximation to the optimal nonlinear filter. One recent advance in this direction is the Wiener chaos decomposition or spectral separation scheme (S^3) for nonlinear filtering in continuous time [17, 18] and similar (related) algorithms that also use the "off-line/on-line separation" idea [19, 20]. A direct numerical approximation to the optimal nonlinear filter is based on computing the convolution integral in the discrete filtering model [15, 29], on using fast solvers for the Fokker-Planck equation in the continuous-discrete filtering model [14, 19, 24], or on solving the Zakai equation in the case of continuous time [6, 8, 9, 10].

Both approaches encounter computational difficulties in practical applications. The problem with EKF and its modifications or, in general, with the assumed density filters, is that they do not work well if the posterior distribution differs from the assumed form. For example, if one assumes a Gaussian distribution and uses EKF while in reality it is multi-peak (far from Gaussian), then the filter completely fails. Assumed density filters (including EKF) fail, for example, in many important situations such as angle-only target-tracking due to divergence, instability, inaccuracy, etc. The reason is that the prespecified density class is too restrictive in the general nonlinear case. Direct approximation is much better in this class of situations but has another important limitation – the "curse of dimensionality". If, for instance, we have a six-dimensional model, which is typical for radar applications, and 100 points are used in each component of the state (in many cases 100 points could even be too few for a satisfactory estimate of the state), then the total number of spatial points is $N = 10^{12}$. If one uses a fast solver with FFT which has complexity $C_d N(\log_2 N)^{d-1}$ for dimension d, then one needs to perform $10^{20}C_6$ flops at each time step, which is simply unacceptable. Even if the Fokker-Planck solver has optimal (linear) complexity (like ADI), we still need to perform $10^{12}C_6$ flops each step. Note that the constant C_d increases with dimension d and $C_1 \geq 8$.

What do we do then? From the one hand to avoid the difficulties that EKF faces, it is desirable to use a direct approximation (rather than an assumed density approach) to the optimal estimate. From the other hand in order to implement direct approximations to high-dimensional real problems, we need (1) to develop fast Fokker-Planck equation solvers with linear complexity, and (2) to reduce the number N of spatial points substantially without reducing the accuracy. The former can be achieved by the so-called operator-splitting method. The latter goal is achieved by reducing adaptively at each time step the size of the spatial domain in which the equation is solved (the domain should be reduced in each direction). This idea leads to the adaptive domain pursuit technique (DPT) that is developed below.

It is worth mentioning that the proposed domain pursuit approach has some similarity with the EKF. In fact, at each time step the DPT tracks a domain (window)[1] in which the target is located and then proceeds the nonlinear filtering in this moving window (or multi-windows), whereas the EKF tracks only two parameters at each time step (the mean and variance of the target state) and then linearizes the nonlinear dynamics around the estimated mean.

The remainder of the report is organized as follows. In Section 2 we outline the basic facts from the theory of optimal nonlinear stochastic filtering for continuous-discrete time model. Section 3 is devoted to the development of the fast algorithm which is based on the convection-diffusion splitting and the domain pursuit technique. In Section 4 we apply the developed general algorithm to a practically important radar target tracking problem in six dimensions. The numerical results obtained in computational experiments are given in Section 4.2. These results demonstrate fairly high accuracy and efficiency of the method and show that it is much more accurate compared to the EKF.

2 Continuous-Discrete Filtering

2.1 Statement of the problem

We are interested in the continuous-discrete filtering model, since this is perhaps the most appropriate model in real target tracking problems. Specifically, the dynamics of target trajectories

[1] This domain may be multiply connected – multi-windows.

is naturally continuous while the observations are usually taken at discrete time moments.

Consider the dynamic system described by the stochastic differential equation

$$\begin{aligned} &dX_t = b(X_t)dt + \sigma(X_t)dW_t, \quad t > 0\,, \\ &X_0 \sim \pi_0\,. \end{aligned} \tag{1}$$

The discrete-time noisy observations are given by

$$Y_k = h_k(X_{t_k}) + R_k(X_{t_k})V_k, \quad k = 0, 1, \dots\,, \tag{2}$$

where $b : \mathbb{R}^d \to \mathbb{R}^d$ is a vector-valued function, $\sigma : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is a matrix with function entries, $\{W_t\}_{t\geq 0}$ is a standard d-dimensional Brownian motion (Wiener process), $t_k = k\tau$ $(\tau > 0)$, π_0 is a (prior) distribution of the initial condition, h_k, R_k are given functions, and V_k are i.i.d. random variables. Without loss of generality, X_0, $\{W_t\}$ and $\{V_k\}$ are assumed to be independent, and the following regularity conditions on the parameters of the model are also assumed: (1) the functions b_k, h_k, Q_k, R_k, and π_0 have bounded derivatives up to an appropriate order, and (2) all the derivatives of π_0 decay at infinity faster than any power of $|x|$.

For simplicity, we only consider the case where the functions b and σ are time-independent. But the discussions that follow can be easily generalized to cover the time-dependent case. In fact, only the coefficients in the Fokker-Planck equation (3) below and the associated semigroup will need to be accordingly modified.

2.2 Fokker-Planck equation

The theory of stochastic differential equations tells us (see [5, 7, 25]) that under certain conditions on b and σ, there exists a unique solution X_t of (1) (in the sense of Ito) and that the probability density $u(t,x)$ of this diffusion process X_t satisfies the Fokker-Planck equation (also known as the Kolmogorov forward equation)

$$\frac{\partial u(t,x)}{\partial t} = \frac{1}{2}\sum_{\mu,\nu=1}^{d} \frac{\partial^2}{\partial x_\mu \partial x_\nu}\Big(a_{\mu\mu}(x)u(t,x)\Big) - \sum_{\nu=1}^{d} \frac{\partial}{\partial x_\nu}\Big(b_\nu(x)u(t,x)\Big) \tag{3}$$

where $a_{\mu\nu}(x)$ is the μ-th row and ν-th column entry of the product matrix $\sigma(x)\sigma(x)^*$, and $b_\nu(x)$ is the ν-th component of the vector $b(x)$.

Let $T(t)$ denote the semigroup associated with the above Fokker-Planck equation. Then its solution $u(t,x)$ with the initial value $u(0,x) = g(x)$ is $[T(t)g](x)$.

We now note that since there is no observation available between t_{k-1} and t_k, the (prior) transition density is the only available information on (t_{k-1}, t_k) and so one has

$$p(X_{t_k} = x \mid \mathbf{Y}^{k-1}) = \left[T(\tau)p(X_{t_{k-1}} = \cdot \mid \mathbf{Y}^{k-1})\right](x).$$

2.3 Unnormalized filtering densities

By using the Bayes formula and the Fokker-Planck equation, one can obtain the conditional densities recursively as follows

$$\begin{aligned} p(X_{t_k} = x \mid \mathbf{Y}^k) &= \frac{1}{c(k)}\alpha_k(x)\left[T(\tau)p(X_{t_{k-1}} = \cdot \mid \mathbf{Y}^{k-1})\right](x), \quad k \geq 1 \\ p(X_0 = x \mid \mathbf{Y}^0) &= \frac{1}{c(0)}\alpha_0(x)\pi_0(x) \end{aligned}$$

where $c(k)$ is the normalizing constant to make the integral of $p(X_{t_k} = x \mid \mathbf{Y}^k)$ to be one, and the "correction term" $\alpha_k(x)$ related to the observation is given by

$$\alpha_k(x) = \exp\left\{-\frac{1}{2}(Y_k - h_k(x))^*(R_k(x)R_k(x)^*)^{-1}(Y_k - h_k(x))\right\}. \tag{4}$$

The calculations are simplified if in place of the usual (normalized) filtering density one uses the *unnormalized filtering densities* (UFD). We define the UFD by[2]

$$\begin{aligned} p_k(x) &= \alpha_k(x)[T(\tau)p_{k-1}](x), \quad k \geq 1 \\ p_0(x) &= \alpha_0(x)\pi_0(x). \end{aligned} \tag{5}$$

It is a standard fact that for any function g such that $I\!\!E|g(X_t)| < \infty$, the conditional expectation of $g(X_{t_k})$ given $\mathbf{Y}^k = \sigma(Y_0, Y_1, \cdots, Y_k)$ can be obtained by

$$I\!\!E(g(X_{t_k}) \mid \mathbf{Y}^k) = \frac{\displaystyle\int_{I\!\!R^d} g(x)p_k(x)dx}{\displaystyle\int_{I\!\!R^d} p_k(x)dx}. \tag{6}$$

This conditional expectation is the best mean-square estimate of $g(X_{t_k})$ if $I\!\!E|g(X_{t_k})|^2 < \infty$. (This is the case if g satisfies $|g(x)| \leq K(1 + |x|^\lambda)$, $\forall x \in I\!\!R^d$, for some λ and $K > 0$; see [16].)

[2] When R_k does not depend on x, our definition here is slightly different from the usually defined UFD. The difference is in $\alpha_k(x)$, where we keep the term $-Y_k^*(R_k R_k^*)^{-1}Y_k/2$. This is done to avoid computational instability: from (4)-(5) we have $\|p_k\|_\infty \leq \|T(\tau)p_{k-1}\|_\infty, \forall k$. In fact, the Fokker-Planck equation for this UFD can also be modified for the purpose of stability. See Section 2.4.3 and Section 2.5.2 of [23] for details.

3 Adaptive Domain Pursuit Method

In this section, we present a windowing technique which is based on the framework of splitting the convection (drift) and diffusion (noise) operators.

3.1 Splitting of convection and diffusion processes

To compute the unnormalized filtering density, a fast Fokker-Planck solver is needed. Our method is based on the operator-splitting technique.

Assuming that in the noise term of (1) the covariance matrix σ is constant and diagonal, the Fokker-Planck equation (3) becomes

$$\frac{\partial u(t,x)}{\partial t} = \frac{1}{2}\sum_{\nu=1}^{n} \frac{\partial^2}{\partial x_\nu^2}\Big(a_\nu u(t,x)\Big) - \sum_{\nu=1}^{n} \frac{\partial}{\partial x_\nu}\Big(b_\nu(x)u(t,x)\Big).$$

To proceed the splitting of convection and diffusion terms, denote by $T_c(t)$ and $T_d(t)$ the solution operators of the equations

$$\frac{\partial v(t,x)}{\partial t} = -\sum_{\nu=1}^{n} \frac{\partial}{\partial x_\nu}\Big(b_\nu(x)v(t,x)\Big),$$

and

$$\frac{\partial w(t,x)}{\partial t} = \frac{1}{2}\sum_{\nu=1}^{n} \frac{\partial^2}{\partial x_\nu}\Big(a_\nu w(t,x)\Big),$$

respectively. Then it can be proved that the following approximation formulae hold (see [23]):

$$\begin{aligned} T(n\tau)\varphi &= (T_d(\tau)T_c(\tau))^n\varphi + O(\tau), && (7)\\ &= \Big(T_c(\frac{\tau}{2})T_d(\tau)T_c(\frac{\tau}{2})\Big)^n\varphi + O(\tau^2). && (8)\end{aligned}$$

Therefore, instead of solving the original Fokker-Planck equation, we only need to solve two simpler equations, for which methods with linear computational complexity exist.

We remark that a big part of computation in solving the two simpler equations can be performed before the observations become available. This pre-calculation substantially speeds up the on-line part of the algorithm.

3.2 Domain pursuit tracker

Even though we have developed optimal solvers for the Fokker-Planck equation, the nonlinear filtering problem can still hardly be solved in real time (especially for large state dimensions). The following natural step is to narrow down the size of the domain in which the Fokker-Planck equation is solved. This can be achieved by a windowing technique based on the convection-diffusion splitting framework.

A numerical approximation to the optimal filter can be expressed in the following form

$$p_k^i = \alpha_k(x^i(k)) \sum_{j \in J} \beta_k^{i,j} p_{k-1}^j, \quad k \geq 1,$$
$$p_0^i = \alpha_0(x^i(0))\pi_0(x^i(0)),$$

where $\{\beta_k^{i,j}\}$ is a (possibly indirect) approximation to the fundamental solution $T(\tau)$ at step k. Note that we use the notation $x^i(k)$ instead of x_k^i for the spatial points because these are vectors in $I\!R^d$ and $x_\nu^i(k)$ will denote the ν-th component.

We remark that in real implementation, the approximate solution $S^h \bar{c}_{k-1}$ (or $\beta_k p_{k-1}$) of the Fokker-Planck equation is not necessarily computed directly from the above summation. For example, if an implicit scheme is used, then S^h contains an inverse matrix which is not inversed directly. Roughly, if the final S^h is sparse, then the summation can be used directly; otherwise the summation will take $O(N^2)$ FLOPS per time step and so some indirect technique such as ADI should be used.

Assume we are given an initial set

$$D_0 = \{x^1(0), \cdots, x^{N_0}(0)\}$$

of "important" points. This set is chosen according to the initial filtering density. For example, these points can be related to the largest values of $p_0(x)$ (or with the most important information on $p_0(x)$). Below we describe how to efficiently compute the "important" points in all the subsequent time steps and also the corresponding values of the unnormalized filtering densities at those points.

If the number N_0 is too large, it can be reduced in the first several filtering steps. Let $K \geq 0$ be a small integer and $\{N_k, k \geq 1\}$ be a decreasing sequence of integers with $N_k = N_K$ for $k > K$, i.e. $N_0 \leq N_1 \leq \cdots \leq N_K = N_{K+1} = \cdots$ Let L be a positive integer. We will first construct an enlarged set of LN_{k-1} candidates for the important points at each step and then choose from them the "best" N_k points according to the correction term α_k.

For simplicity, we assume again that the covariance matrix σ is constant and diagonal. Our algorithm can be described as follows.

The DPT Algorithm

- Initialization: start with time $k = 0$, domain D_0 of size N_0, and $p_0^i = p_0(x^i(0)), 1 \le i \le N_0$.
- Iteration: for $k \le K$, run algorithm DPT(k) (see below) with reduction of domain size N_k.
- For $k > K$, run algorithm DPT(k) with moving domain D_k of fixed size $N_k = N_{k-1}$.

The algorithm DPT(k) proceeds as follows

(1) for $i = 1, \cdots, N_{k-1}$, solve the stochastic differential/integral equation

$$X_t = x^i(k-1) + \int_0^t b(X_s)ds + \int_0^t \sigma dW_s, \quad t \in (0, \Delta t] \,,$$

with L different sample paths of the Wiener process W_t, including the trivial case $W_t \equiv 0$, and denote the solution X_τ at time $t = \tau$ with the j-th sample path of W_t by $\xi^{i,j}$, $j = 1, \cdots, L$;

(2) determine the set D_k of N_k important points $x^1(k), \cdots, x^{N_k}(k)$ as those $\xi^{i,j}$ with the largest values of $\alpha_k(\xi^{i,j})$ for all i and j, or, for each i, $x^i(k)$ has the largest value of $\alpha_k(\xi^{i,j})$ for $j = 1, \cdots, L$;

(3) for $i = 1, \cdots, N_k$, compute

$$p_k^i = \alpha_k(x^i(k)) \sum_{j \in J_k^i} \beta_k^{i,j} p_{k-1}^j \,,$$

where $\beta_k^{i,j}$ is computed according to one of the following two formulae which follow from the operator-splitting schemes discussed in the previous section:

$$\beta_k^{i,j} = \exp\left\{ -\sum_{\nu=1}^{d} \frac{(x_\nu^i(k) - x_\nu^j(k-1) - b_\nu(x^j(k-1))\tau)^2}{2\sigma_\nu^2 \tau} \right\} ,$$

or

$$\beta_k^{i,j} = \exp\left\{ -\sum_{\nu=1}^{d} \frac{(x_\nu^i(k) - x_\nu^j(k-1))^2}{2\sigma_\nu^2 \tau} - (\nabla \cdot b)\Big(\frac{x^i(k) + x^j(k-1)}{2}\Big) \right\} ,$$

and $J_k^i = \{j : 1 \le j \le N_{k-1}, \min(\beta_k^{i,j}, p_{k-1}^j) > \tau\}$, τ being a thresholding tolerance.

We stress that in general the domains D_k may be multiply connected, i.e. may contain multiple windows (pieces). The block diagram of the domain pursuit tracker is shown in Figure 1.

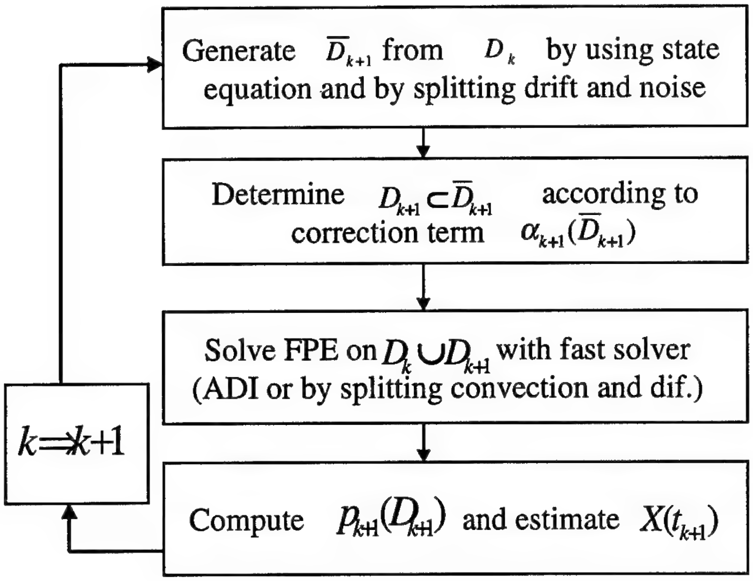
Generate $\overline{D}_{k+1}$ from D_k by using state equation and by splitting drift and noise
Determine $D_{k+1} \subset \overline{D}_{k+1}$ according to correction term $\alpha_{k+1}(\overline{D}_{k+1})$
Solve FPE on $D_k \cup D_{k+1}$ with fast solver (ADI or by splitting convection and dif.)
$k \Rightarrow k+1$
Compute $p_{k+1}(D_{k+1})$ and estimate $X(t_{k+1})$

Figure 1: Domain pursuit tracker

The remarkable feature of the proposed filtering algorithm is that the domain of interest adaptively changes and, as a result, the number of spatial points in the domain is usually reduced to a relatively small number. Hence the computational complexity is reduced tremendously when compared with the case where a fixed size domain is used for the whole computation. In general, the number N of spatial points is exponential in d, i.e., $N \sim n^d$, but we are reducing this number for each n of the d dimensions. Therefore, the computational cost is exponentially reduced in our algorithms in high dimensions.

4 Application to the Problem of Ballistic Target Tracking

4.1 The tracking problem

To illustrate the performance of the proposed algorithm, let us consider a real RADAR tracking problem, which is well-known to be difficult despite it does not contain perturbations in dynamics. (In principle we can also handle a similar problem with infra-red or other kinds of angle-only measurements and the model that would include dynamics noise.)

There is a radar with geodetic latitude θ, longitude λ, and height h, observing a ballistic missile and generating radar range R, azimuth A, and elevation E every τ seconds (see Figure 2).

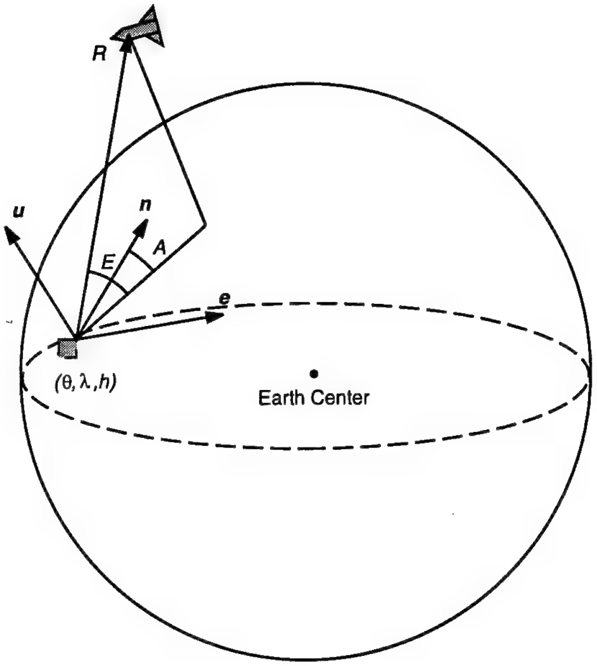

Figure 2: Radar geometry

The missile is assumed to be in unpowered ballistic flight whose six dimensional dynamic equations of motion are

$$\frac{dX(t)}{dt} = V(t) \,, \qquad \frac{dV(t)}{dt} = -\mu \frac{X(t)}{\|X(t)\|^3} \,, \qquad t \geq 0 \,, \tag{9}$$

where $\mu = 3.986012 \times 10^{14}$, $X(t) = (X_1(t), X_2(t), X_3(t))$ and $V(t) = (V_1(t), V_2(t), V_3(t))$ are the position and velocity of the missile at time moment t, and $X(0)$ and $V(0)$ are Gaussian random vectors with known mean

$$\begin{aligned} I\!\!E(X(0)) &= [0, 0, 7.45005724 \times 10^6]^T(\mathrm{m}), \\ I\!\!E(V(0)) &= [-3.96745 \times 10^3, -2.37208 \times 10^3, 2.15685 \times 10^3]^T(\mathrm{m/sec}), \end{aligned}$$

and covariance

$$\mathrm{Cov}(X(0), V(0)) = \mathrm{diag}[4 \times 10^6, 4 \times 10^6, 4 \times 10^6, 10^4/3, 10^4/3, 10^4/3] \,.$$

The computation of the radar measured range, azimuth, and elevation from the missile's true inertial position at time $t_k = k\tau$ proceeds as follows:

$$\begin{aligned} R(k) &= \|X_L(t_k)\| + \varepsilon_1 v_1(k)\,, \\ A(k) &= \tan^{-1}\frac{X_{L,1}(t_k)}{X_{L,2}(t_k)} + \varepsilon_2 v_2(k)\,, \\ E(k) &= \sin^{-1}\frac{X_{L,3}(t_k)}{\|X_L(t_k)\|} + \varepsilon_3 v_3(k)\,, \end{aligned} \tag{10}$$

where $\varepsilon_1 = 16,\ \varepsilon_2 = \varepsilon_3 = \sqrt{3.04617 \times 10^{-6}}$, and

$$X_L(t) = T_{\text{LOC,ECEF}} \cdot [T_{\text{ECEF,ECI}}(t) \cdot X(t) - X_S]\,.$$

Here LOC, ECEF, and ECI stand for the three relevant coordinate systems: ECI for Earth Centered Inertial True-of-Date, ECEF for Earth Centered Earth Fixed, and LOC for Local East-North-Up. The coordinate transformation matrices $T_{\text{LOC,ECEF}}$ and $T_{\text{ECEF,ECI}}(t)$ and the radar site location X_S are given by

$$T_{\text{LOC,ECEF}} = \begin{bmatrix} -\sin\lambda & -\sin\theta\cos\lambda & \cos\theta\cos\lambda \\ \cos\lambda & -\sin\theta\sin\lambda & \cos\theta\sin\lambda \\ 0 & \cos\theta & \sin\theta \end{bmatrix},$$

$$T_{\text{ECEF,ECI}}(t) = \begin{bmatrix} \cos(\omega_e t) & \sin(\omega_e t) & 0 \\ -\sin(\omega_e t) & \cos(\omega_e t) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$X_S = \begin{bmatrix} (\alpha + h)\cos\theta\cos\lambda \\ (\alpha + h)\cos\theta\sin\lambda \\ (\beta + h)\sin\theta \end{bmatrix}, \quad \alpha = \frac{a_e}{\sqrt{1 - e^2\sin^2\theta}}, \quad \beta = \frac{a_e(1 - e^2)}{\sqrt{1 - e^2\sin^2\theta}},$$

where $\theta = 1.12032684685$ (rad), $\lambda = -2.60246044764$ (rad), $\omega_e = 7.2722052162296 \times 10^{-5}$(rad/sec), $a_e = 6.37815 \times 10^6$ (m), $e^2 = 6.69342162296 \times 10^{-3}$, $h = 0$ (m).

4.2 Monte Carlo simulations

Below we present the results of computational experiments. ($L = 1$ was taken for Algorithm 1.)

Assume the observations are available at every $\tau = 1$ second. Since there is no noise in the state dynamics (9), we do not need the diffusion smoothing as described for the general model in the algorithm. If the noise is added in the dynamics, then the general algorithm should be applied.

In our simulation, we ran our filter 200 times with random initial conditions and random observations. We used two different values of N, the number of spatial points in the moving domain

Table 1: Computational complexity of the algorithm

Simulation	N	flops	cpu1	cpu2
Simulation 1	729	72,182	0.14	0.04
Simulation 2	15,625	1,546,886	0.64	0.48

(window). Specifically, $N = 3^6 = 729$ and $N = 5^6 = 15,625$. The average computational cost for the two experiments in terms of CPU seconds and FLOPS per time step are given in Table 1, where cpu1 and cpu2 are the CPU time for one step of calculations with and without graphics, respectively. The computation was performed with Matlab on a Sun Ultra Enterprise 4000 at the University of Southern California. From this table, it is clear that real-time performance has been achieved even for tracking 10 to 20 targets.

For both values of N, we tested three different situations. In the first case we assume that the true initial location of the target and its velocity are exactly on the grid; in the second case it is assumed only that the initial velocity is exactly on the grid; and in the third case both the initial location and velocity are not on the grid (the most realistic case). Of course, in either case, the filter does not know the true initial location of the target. In the first case performance is perfect: with $N = 729$, the average errors for both $X(t)$ and $V(t)$ became close to zero after $t = 140$ seconds. The average errors, defined as

$$X\mathrm{err},_\nu(t) = \frac{1}{200}\sum_{n=1}^{200} |X_\nu^{(n)}(t) - \hat{X}_\nu^{(n)}(t)| \quad \text{and} \quad V\mathrm{err},_\nu(t) = \frac{1}{200}\sum_{n=1}^{200} |V_\nu^{(n)}(t) - \hat{V}_\nu^{(n)}(t)|$$

($\nu = 1, 2, 3$) are shown in Figure 4 (for the two simulations). Note that because the initial error in $X(0)$ is too large, it is not shown in the picture. See Figure 3 for the average errors (including the initial errors) according to the state dynamics without any observed information.

In the second and third cases, when the initial variance (for the true initial state) is relatively small, we also obtained good results. For example, in the second case, with the initial variance

$$\mathrm{Cov}(X(0), V(0)) = \mathrm{diag}[10^4, 10^4, 10^4, 10, 10, 10]\,,$$

the average errors in V go to zero and the average errors in X begin to decrease after $t = 90$ seconds. These errors are shown in Figure 5. Again the initial error in $X(0)$ is not shown in the picture. In the third case, with the initial variance

$$\mathrm{Cov}(X(0), V(0)) = \mathrm{diag}[10^4, 10^4, 10^4, 1, 1, 1]\,,$$

the average errors behave similarly except that they decrease at a slower rate.

The real initial errors used in experiments are listed in Table 2. Typical errors at $t = 100$ seconds (with the original large error) are shown in Table 3. This table contains data for DPT and EKF. It may be seen that the proposed method is much more accurate.

Table 2: Real initial errors

Variance	$X_{\text{err},1}(0)$	$X_{\text{err},2}(0)$	$X_{\text{err},3}(0)$	$V_{\text{err},1}(0)$	$V_{\text{err},2}(0)$	$V_{\text{err},3}(0)$
Initial Errors	1509.2912	1565.5626	1608.2539	43.0280	50.5172	43.7596

Table 3: Comparison with EKF

Method	$\|X_{\text{err}}(100)\|$	$\|V_{\text{err}}(100)\|$
EKF	100 (m)	0.5 (m/sec)
DPT	<15 (m)	0.1 (m/sec)
Ratio	6.7	5.0

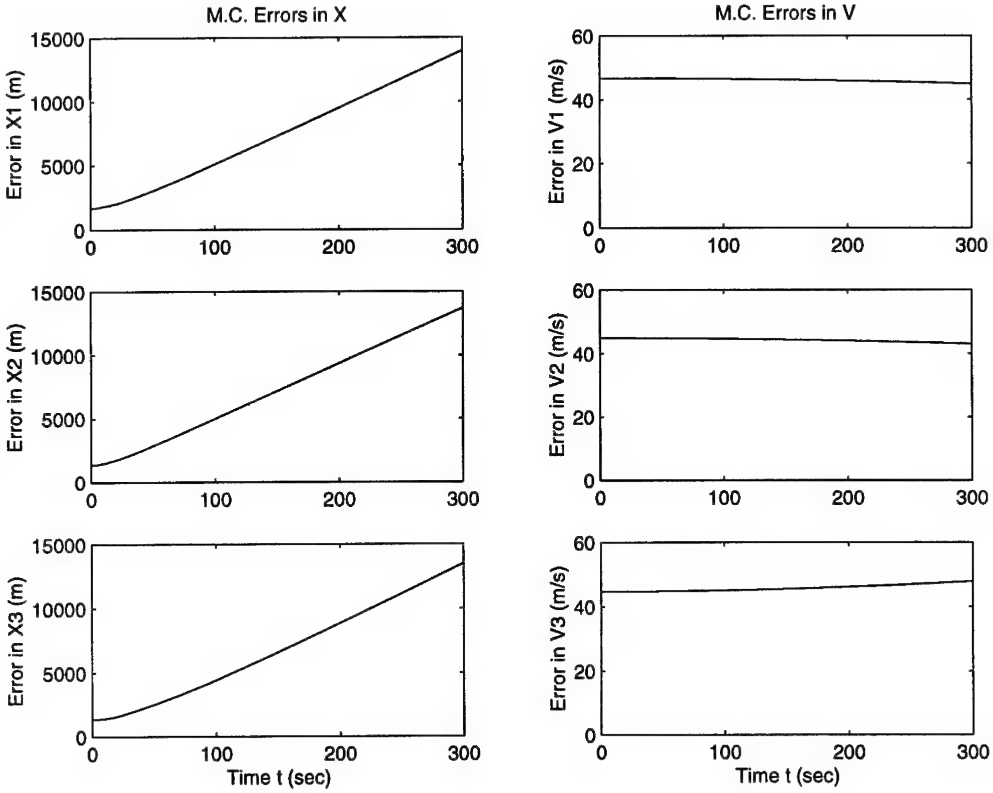

Figure 3: Errors without observations

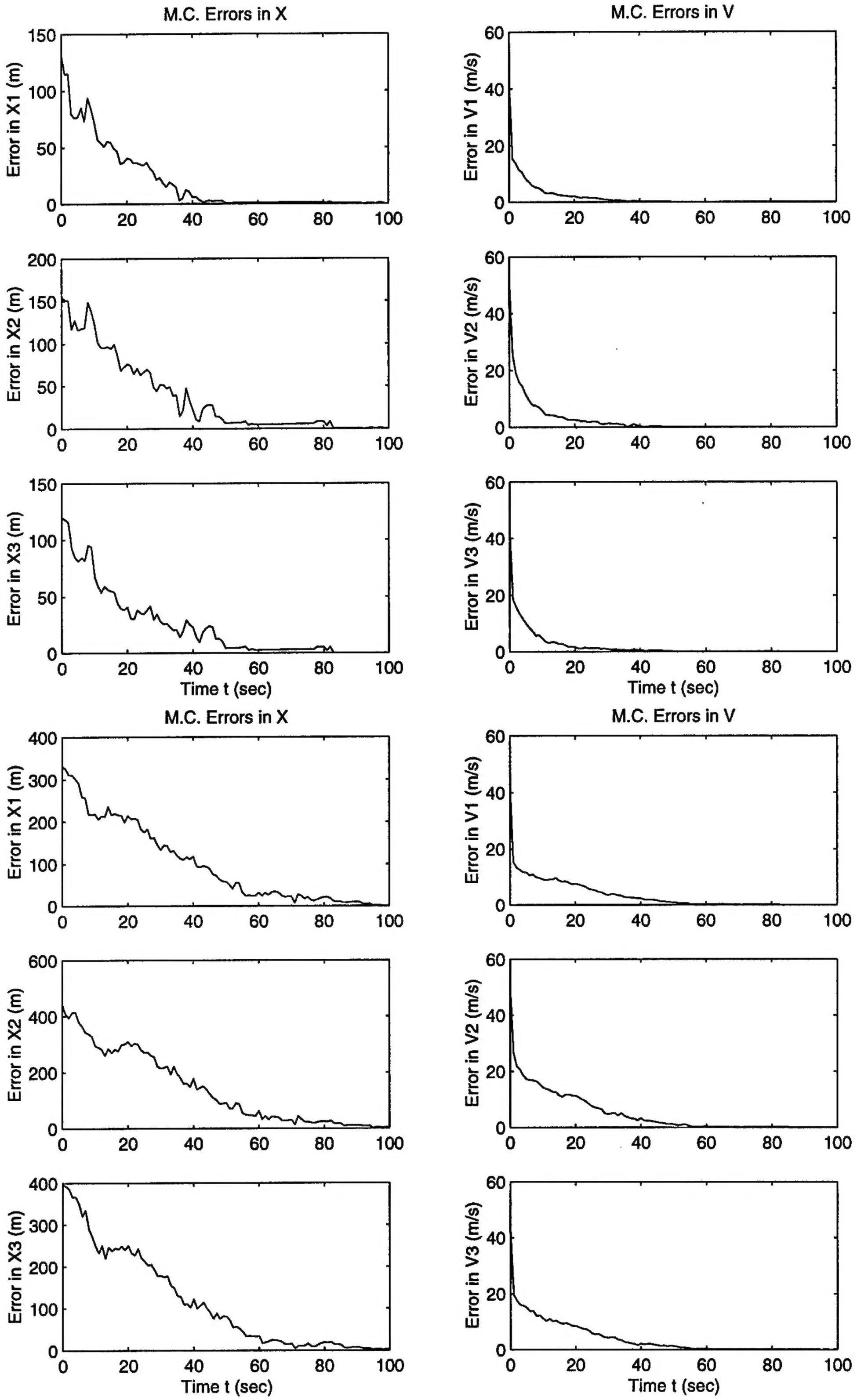

Figure 4: Errors in Case 1. Left – simulation 1 ($N = 3^6$); Right – simulation 2 ($N = 5^6$)

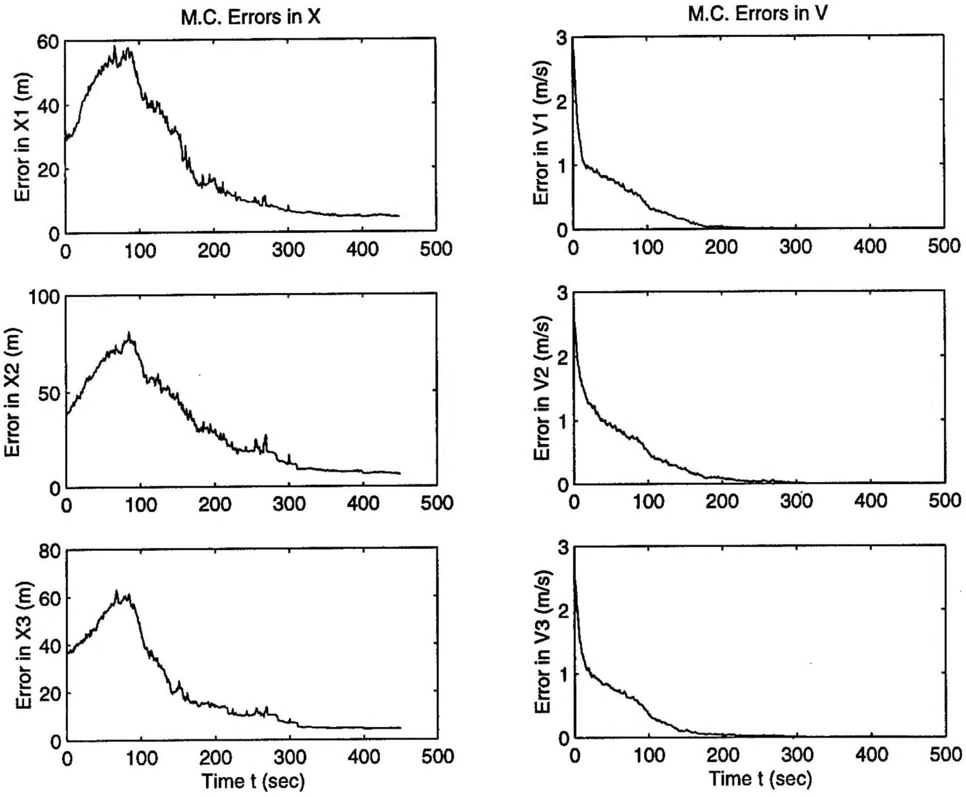

Figure 5: Errors in Case 2, $N = 3^6$

5 Conclusion

1. In this report we described the developed nonlinear filtering algorithm that is based on the "domain pursuit method". This method is directed towards obtaining robust nonlinear tracking algorithms with manageable complexity and high statistical performance (close to the optimal level).

2. The algorithm is applied to a realistic problem that is typical for tracking ballistic missiles by radar. The considered scenario includes targets with "hard" trajectories that should be localized in 100-150 seconds. The results of simulation show that the developed algorithm substantially outperforms the conventional EKF tracker it terms of mean-squared tracking error and at the same time has satisfactory computational complexity (may be applied in real time).

6 Acknowledgement

We are grateful to Roy Danchick of TRW for useful discussions and for providing the data and realistic parameters for analysis.

References

[1] R. L. Bellaire, E. W. Kamen, and S. M. Zabin. A new nonlinear iterated filter with applications to target tracking. In *SPIE Proc. Signal & Data Processing of Small Targets*, San Diego, 1995.

[2] A. Bensoussan. *Stochastic Control of Partially Observable Systems.* Cambridge University Press, 1992.

[3] D. Brigo, B. Hanzon, and F. LeGland. A differential geometric approach to nonlinear filtering: the projection filter. Publication Interne No. 914, IRISA, 1995.

[4] R. J. Elliott, L. Aggoun, and J. B. Moore. *Hidden Markov Models: Estimation and Control.* Springer-Verlag, New York, 1995.

[5] A. Friedman. *Stochastic Differential Equations and Applications.* Academic Press, 1975.

[6] C.-P. Fung. *New Numerical Algorithms for Nonlinear Filtering.* PhD thesis, University of Southern California, Los Angeles, November 1995.

[7] C. W. Gardiner. *Handbook of Stochastic Methods for Physics, Chemistry, and the Natural Sciences.* Springer-Verlag, Berlin, 2nd edition, 1985.

[8] A. Germani and M. Piccioni. Semi-discretisation of stochastic partial differential equations on $\mathbb{R}^d$ by a finite element technique. *Stochastics*, 23(2):131–148, 1988.

[9] K. Ito. Approximation of the Zakai equation for nonlinear filtering. *SIAM J. Control Optim.*, 34(2):620–634, 1996.

[10] K. Ito and B. L. Rozovskii. Approximation of the Kushner equation for nonlinear filtering. Preprint, 1998.

[11] A. H. Jazwinski. *Stochastic Processes and Filtering Theory.* Academic Press, New York, 1970.

[12] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte. A new approach to filtering nonlinear systems. In *Proc. of the American Control Conf.*, pages 1628–1632, Seattle, WA, June 1995.

[13] G. Kallianpur. *Stochastic Filtering Theory.* Springer-Verlag, New York, 1980.

[14] K. Kastella and A. Zatezalo. A nonlinear filter for real-time joint tracking and recognition. IMA Preprint Ser. 958, Inst. Math. Its Appl., University of Minnesota, Minneapolis, 1998.

[15] G. Kitagawa and W. Gersch. *Smoothness Priors Analysis of Time Series.* Springer, 1996.

[16] R. S. Liptser and A. N. Shiryayev. *Statistics of Random Processes I: General Theory.* Springer, 1977.

[17] S. V. Lototsky. *Problems in Statistics of Stochastic Differential Equations.* PhD thesis, University of Southern California, Los Angeles, August 1996.

[18] S. V. Lototsky, R. Mikulevicius, and B. L. Rozovskii. Nonlinear filtering revisited: a spectral approach. *SIAM J. Control Optim.*, 35(2):435–461, March 1997.

[19] S. V. Lototsky, C. Rao, and B. L. Rozovskii. Fast nonlinear filter for continuous-discrete time multiple models. In *Proc. 35th Conf. Decision and Control*, pages 4060–4064, 1996.

[20] S. V. Lototsky and B. L. Rozovskii. Recursive nonlinear filter for a continuous-discrete time model: separation of parameters and observations. *IEEE Trans. Autom. Control*, 1997.

[21] P. S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 2. Academic Press, 1979.

[22] E. Pardoux. Filtrage non linéaire et équations aux dérivées partielles stochastiques associées. In P.L. Hennequin, editor, *Ecole d'Eté de Probabilités de Saint-Flour XIX – 1989*, volume 1464 of *Lecture Notes in Mathematics*, pages 69–163. Springer-Verlag, Berlin, 1991.

[23] C. Rao. *Nonlinear Filtering and Evolution Equations: Fast Algorithms with Applications to Target Tracking.* PhD thesis, University of Southern California, Los Angeles, August 1998.

[24] C. Rao and B. L. Rozovskii. A fast filter for nonlinear systems with discrete observations. *Computing Science and Statistics*, 29(2):619–625, 1997.

[25] H. Risken. *The Fokker-Planck Equation: Methods of Solution and Applications.* Springer-Verlag, Berlin, 2nd edition, 1989.

[26] B. L. Rozovskii. *Stochastic Evolution Systems: Linear Theory and Applications to Non-linear Filtering.* Kluwer Academic Publishers, Dordrecht, 1990.

[27] H. W. Sorenson and D. L. Alspach. Recursive Bayesian estimation using Gaussian sums. *Automatica—J. IFAC*, 7:465–479, 1971.

[28] R. L. Stratonovich. *Conditional Markov Processes and Their Application to the Theory of Optimal Control.* American Elsevier Publishing Co., Inc., New York, 1968.

[29] H. Tanizaki. *Nonlinear Filters: Estimation and Applications.* Springer-Verlag, Berlin, 2nd edition, 1996.